In response to the Office Action of Feb. 9, 2006, please reconsider the pending claims based on the following amendment.

Amendments to the Claims

1. (original) A method for bridging information from a pre-boot environment to a runtime environment, comprising:

registering an event logging handler with a plurality of event handlers in a pre-boot environment;

storing a plurality of event data in a memory-based buffer, by the event logging handler, as events occur;

retrieving the plurality of event data from the memory-based buffer; and

storing the retrieved plurality of event data in a memory location accessible by an operating system, the storing being performed prior to launching of the operating system.

2. (currently amended) The method as recited in claim 1, wherein the event data stored in the memory-based buffer shadows event data stored in a proprietary memory, wherein the proprietary memory is accessible at pre-boot, but not by the operating system.

3. (original) The method as recited in claim 1, wherein the proprietary memory is a non-volatile memory.

4. (original) The method as recited in claim 1, wherein the memory location accessible by the operating system is identified by a globally unique identifier (GUID) pointer.

5. (original) The method as recited in claim 4, wherein the GUID pointer points to a memory location reserved for an extensible firmware interface (EFI) configuration table.

6. (original) The method as recited in claim 5, wherein buffer space in the EFI configuration table is made available for storing the memory-based buffer after all pre-boot events have been logged.

7. (original) The method as recited in claim 4, wherein the GUID pointer points to a memory location reserved for an advanced configuration power interface (ACPI) configuration table.

8. (original) The method as recited in claim 7, wherein buffer space in the ACPI configuration table is made available for storing the memory-based buffer after all pre-boot events have been logged.

9. (original) The method as recited in claim 1, wherein event data comprises: progress data, status data, error logging information, and general system information.

10. (original) The method as recited in claim 1, further comprising accessing the memory-based buffer by an operating system agent.

11. (original) The method as recited in claim 10, further comprising performing an action, by the operating system agent, in response to data accessed from the memory-based buffer.

12. (original) The method as recited in claim 11, wherein the action performed is selected from the group of actions consisting of generating an alert, displaying event data, reporting event data, saving event data, transmitting event data, and transmitting a notification to a user.

13. (original) The method as recited in claim 12, wherein displaying event data comprises:

generating one or more eXtensible Markup Language (XML) pages having event data in a standardized format; and

displaying the one or more XML pages using a standards based browser.

14. (original) The method as recited in claim 13, wherein the displaying occurs during runtime.

15. (original) A method for displaying pre-boot event data in a runtime environment, comprising:

retrieving event data from a memory-based buffer, by an operating system agent during runtime, the memory-based buffer being generated in a pre-boot environment by an event logging handler, wherein the memory-based buffer resides in a reserved portion of system memory known to both the pre-boot environment and the runtime environment; and

displaying the event data during runtime, by the operating system agent.

16. (original) The method as recited by claim 15, wherein the memory-based buffer is identified by a globally unique identifier (GUID) pointer.

17. (original) The method as recited by claim 16, wherein the GUID pointer points to a location in memory corresponding to an extensible firmware interface (EFI) configuration table.

18. (original) The method as recited by claim 16, wherein the GUID pointer points to a location in memory corresponding to an advanced configuration power interface (ACPI) configuration table.

19. (original) The method as recited by claim 15, wherein displaying comprises:

retrieving event data from the memory-based buffer;

generating at least one page for display, based upon the retrieved event data; and

displaying the at least one generated page on a display device.

20. (original) The method as recited by claim 19, wherein the generated pages are in extensible markup language (XML) format.

21. (original) The method as recited by claim 19, wherein the at least one generated page are displayed using a standards-based browser.

22. (original) An article of manufacture comprising a machine-accessible medium having instructions that, when executed, cause the machine to:

register an event logging handler with a plurality of event handlers in a pre-boot environment;

store a plurality of event data in a memory-based buffer, by the event logging handler, as events occur;

retrieve the plurality of event data from the memory-based buffer; and

store the retrieved plurality of event data in a memory location accessible by an operating system, the storing being performed prior to launching of the operating system.

23. (currently amended) The article as recited in claim 22, wherein the event data stored in the memory-based buffer shadows event data stored in a proprietary non-volatile memory, wherein the non-volatile memory is accessible at pre-boot, but not by the operating system.

24. (original) The article as recited in claim 22, wherein the memory location accessible by the operating system is identified by a globally unique identifier (GUID) pointer.

25. (original) The article as recited in claim 24, wherein the GUID pointer points to a memory location reserved for an extensible firmware interface (EFI) configuration table.

26. (original) The article as recited in claim 25, wherein buffer space in the EFI configuration table is made available for storing the memory-based buffer after all pre-boot events have been logged.

27. (original) The article as recited in claim 24, wherein the GUID pointer points to a memory location reserved for an advanced configuration power interface (ACPI) configuration table.

28. (original) The article as recited in claim 27, wherein buffer space in the ACPI configuration table is made available for storing the memory-based buffer after all pre-boot events have been logged.

29. (original) The article as recited in claim 22, wherein event data comprises: progress data, status data, error logging information, and general system information.

30. (original) The article as recited in claim 22, further comprising instructions that cause the machine to access the memory-based buffer by an operating system agent.

31. (original) The article as recited in claim 30, further comprising instructions that cause the machine to perform an action, by the operating system agent, in response to data accessed from the memory-based buffer.

32. (original) The article as recited in claim 31, wherein the action performed is selected from the group of actions consisting of generating an alert, displaying event data, reporting event data, saving event data, transmitting event data, and transmitting a notification to a user.

33. (original) The article as recited in claim 32, wherein displaying event data comprises:

generating one or more eXtensible Markup Language (XML) pages having event data in a standardized format; and

displaying the one or more XML pages using a standards based browser.

34. (original) The article as recited in claim 33, wherein the displaying occurs during runtime.

35. (original) An article of manufacture comprising a machine-accessible medium, having instructions that, when executed, cause the machine to:

retrieve event data from a memory-based buffer, by an operating system agent during runtime, the memory-based buffer being generated in a pre-boot environment by an event logging handler, wherein the memory-based buffer resides in a reserved portion of system memory known to both the pre-boot environment and the runtime environment; and

display the event data during runtime, by the operating system agent.

36. (original) The article as recited by claim 35, wherein the memory-based buffer is identified by a globally unique identifier (GUID) pointer.

37. (original) The article as recited by claim 36, wherein the GUID pointer points to a location in memory corresponding to an extensible firmware interface (EFI) configuration table.

38. (original) The article as recited by claim 36, wherein the GUID pointer points to a location in memory corresponding to an advanced configuration power interface (ACPI) configuration table.

39. (original) The article as recited by claim 35, wherein displaying comprises:

retrieving event data from the memory-based buffer;

generating at least one page for display, based upon the retrieved event data; and

displaying the at least one generated page on a display device.

40. (original) The article as recited by claim 39, wherein the generated pages are in extensible markup language (XML) format.

41. (original) The article as recited by claim 39, wherein the at least one generated page is displayed using a standards based browser.

42. (currently amended) A system for bridging information from a pre-boot environment to a runtime environment, comprising:

a processor, wherein the processor is operatively coupled with a non-volatile pre-boot memory store and a random access memory (RAM);

a memory for storing event data in a pre-boot environment operatively coupled with the processor; a memory-based buffer for shadowing pre-boot environment event data, the memory-based buffer being accessible by an operating system agent, and operatively coupled to the processor; and

an event logging handler running on the processor during pre-boot, the event logging handler for registering the pre-boot event data to the memory-based buffer, wherein the event logging handler is to be registered with a plurality of event handlers in the pre-boot environment.

43. (original) The system as recited in claim 42, wherein the pre-boot memory store is flash RAM storing a basic input/output system (BIOS).

44. (original) The system as recited in claim 42, wherein the memory-based buffer is stored in an extensible firmware interface (EFI) configuration table, and is accessible using a globally unique identifier (GUID) pointer.

45. (original) The system as recited in claim 42, wherein the memory-based buffer is stored in an advanced configuration power interface (ACPI) configuration table, and is accessible using a globally unique identifier (GUID) pointer.

46. (original) The system as recited in claim 42, further comprising:

a display device operatively coupled to the processor and in communication with the operating system agent; and

a user interface, the user interface retrieving the pre-boot environment event data from the memory-based store and formatting the pre-boot environment event data for displaying on the display device.

47. (original) The system as recited in claim 46, wherein the pre-boot environment event data is formatted into a series of extensible markup language (XML) pages by the user interface.

48. (original) The system as recited in claim 42, wherein the memory for storing event data in a pre-boot environment is flash RAM.

49. (original) The system as recited in claim 48, wherein the memory for storing event data resides on the flash RAM comprising the non-volatile pre-boot memory store.

50. (original) The system as recited in claim 48, wherein the memory for storing event data is a firmware hub comprising a BIOS and vital product data (VPD).